



# Bandit-based Estimation of Distribution Algorithms for Noisy Optimization: Rigorous Runtime Analysis

Philippe Rolet, Olivier Teytaud

## ► To cite this version:

Philippe Rolet, Olivier Teytaud. Bandit-based Estimation of Distribution Algorithms for Noisy Optimization: Rigorous Runtime Analysis. Lion4, 2010, Venice, Italy. inria-00437140

**HAL Id: inria-00437140**

**<https://inria.hal.science/inria-00437140>**

Submitted on 29 Nov 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Bandit-based Estimation of Distribution Algorithms for Noisy Optimization: Rigorous Runtime Analysis

P. Rolet, O. Teytaud

TAO (Inria), LRI, UMR 8623 (CNRS - Univ. Paris-Sud),  
bat 490 Univ. Paris-Sud 91405 Orsay, France

**Abstract.** We show complexity bounds for noisy optimization, in frameworks in which noise is stronger than in previously published papers[19]. We also propose an algorithm based on bandits (variants of [16]) that reaches the bound within logarithmic factors. We emphasize the differences with empirical derived published algorithms.

## 1 Introduction

Optimization in noisy environments deals with fitness functions for which multiple evaluations do not necessarily give the same result, due to noise in the computation. Noise occurs in many real world applications. In numerous cases, it is due to finite precision of Monte-Carlo sampling, and often the noise is centered to zero, *i.e.* the goal is to have an optimal expected value.

The simplest solution for reducing noise consists in evaluating several times the same point; by averaging, the variance decreases. Unfortunately, depending on detailed experimental conditions, various points of views can be found in the literature, notably regarding evolutionary strategies.

[12] concludes in the case of genetic algorithms that averaging does not perform well: when considering the tradeoff between the computational overhead of averaging and the convergence rate, convergences are always slower with averaging. In the case of evolution strategies (see [8]), [14] and [1] draw different conclusions: [14] concludes that strong averaging is required, whereas [1] concludes that increasing the population size is better than averaging. According to [1], results in [14] are due to a poor mutation strength adaptation schema; however, interestingly, they point out that for various noise models, each usual mutation-strength adaptation schema can lead to poor results: [1] compares mutative self-adaptation [25, 26] (two variants, one with arithmetic averaging of mutation strength and the other with geometric averaging) and cumulative self-adaptation [15], and concludes that in both cases, there are simple examples of noise models for which poor behaviors can occur. It might then be that averaging is required to avoid such behaviours: this problem remains open. An improvement of cumulative step-length adaptation is possible with increased population size, but only to a limited extent [2]. An analysis using Markov chains has been

adapted from the noise-free case [9, 5] to the noisy case in [20, 27]; reevaluations are suggested, but it is pointed out that in many cases, this would not be sufficient.

[17] proposed the use of “bandits” [21, 4] and Bernstein’s races [24] for choosing the number of function evaluations spent on a given individual (bandits and races will be presented in section 3). This idea, albeit promising, could not be applied as such to the problem; the authors had to add several tricks in the implementation in order to get acceptable results. Furthermore, they could not get convergence rates, convergence proofs, or convincing experimental curves exposing a good behavior. The main goal of this paper is to get convergence proofs and convergence rates using a modified bandit algorithm; the difference between our approach and the approach in [17] will be clearly emphasized in section 4.

An peculiar case of the noisy optimization setting is when the variance of the noise decreases to zero near the optimum. This has been tackled in [19], when the noise satisfies conditions such as, typically,

$$f_{noise}(x) = f(x) \times (1 + N)$$

where  $f(x)$  is the sphere function  $f(x) = \|x\|^2$  and  $N$  is an independent noise variable. Assuming that  $N$  has a density in a bounded range  $[m, M]$  with  $-1 < m < M$ , [19] shows that the scale-invariant  $1 + 1$ -ES converges. However, there is no information on the convergence rate, and the theoretical algorithm cannot be used in practice (see note 5 in [19]) as it uses some prior information, at each iteration, on the distance to the optimum. In this paper:

- we derive a lower bound on the number of requests necessary for reaching a given precision for a given confidence level;
- we consider a different model of noise, including cases in which the noisy measurement can be arbitrarily small with respect to the expected fitness, i.e.  $f_{noise}(x)/f(x)$  is not lower bounded;
- we propose an *explicit* algorithm: it does not require information on the position of the optimum, or on the distance towards the optimum;
- we prove a tight convergence rate for any fixed dimension.

In all the paper,  $\#E$  denotes the cardinal of the set  $E$ ,  $(x)_i$  denotes the  $i^{th}$  coordinate of vector  $x$ , and  $[[a, b]]$  denotes  $\{a, a + 1, a + 2, \dots, b\}$ .

The paper is organized as follows. Section 2 will show a lower bound on the number of requests necessary for reaching a given precision with a given probability. Section 3 will propose an algorithm for noisy optimization, along with a proof that this algorithm matches the lower bound. This shows the tightness of the lower bound given in section 2, within logarithmic factors. Section 4 will then discuss the difference with existing algorithms and emphasize the importance of subtle elements in the generation of the offspring.

## 2 Lower bounds

In this section, we show a lower bound on the number of requests before an optimization algorithm can find the optimum with precision  $\epsilon$  and confidence

$1 - \delta$ . This means that it is not possible to design an algorithm which finds, with a smaller number of requests to the fitness function, the optimum of the fitness function with precision  $\epsilon$  and with probability  $1 - \delta$ .

The optimization framework is described in Algorithm 1. This is a black-box optimization framework: the algorithm can request the fitness value at any point of the domain, and this costs one request. No other information on the fitness function is available.

We consider a fitness function  $f$  parameterized by the (unknown) location of its optimum,  $t$ . The goal is to find the optimum  $t$  of  $f(., t)$ , by observing noisy measurement of  $f$  at  $x_i$ : 1 with probability  $1 - f(x_i, t)$  (i.e. if  $\theta_i > f(x_i, t)$ ) and 0 otherwise.  $\theta$  is the random variable accounting for the noise. This fits applications based on highly noisy optimization, such as games: let  $x$  be a parameter of a game strategy, that we wish to set at its best value; one noisy observation is a game against a baseline, resulting either in a win or in a loss; the aim is to find the value of  $x$  maximizing the probability of winning.

In the following,  $t$  is not handled stochastically, i.e. the lower bounds are not computed in expectation w.r.t all the possible fitness functions yielded by different values of  $t$ . Rather, we will consider the worst case on  $t$ . Therefore the only random variable in this framework is  $\theta$ , which decides the answers (see Algorithm 1). As a consequence, all probability / expectation operators are w.r.t  $\theta$ .

---

**Algorithm 1** Noisy optimization framework. *Opt* is an optimization algorithm; it takes as input a sequence of visited points and their measured fitness values, and outputs a new point to be visited. Fitness values are binary: we are looking for points  $x$  of the domain such that  $f(x, t)$  is as small as possible. The algorithm *Opt* is successful on target function  $f$  parameterized by  $t$  and random noise  $\theta$  if  $Reward(t, \theta, Opt)$  is small; details in the text.

---

Parameters:  $N$ , number of fitness evaluations;  $t$ , unknown element of  $X$ .  
 $\theta$ : random state of the nature  $\in [0, 1]^N$ ; each coordinate  $\theta_i$  for  $i \in \{1, 2, \dots\}$  is uniformly distributed in  $[0, 1]$ .

```

for  $n \in [[0, N - 1]]$  do
   $x_{n+1}^{t, \theta} = Opt(x_1^{t, \theta}, \dots, x_n^{t, \theta}, y_1^{t, \theta}, \dots, y_n^{t, \theta})$ 
  if  $n < N - 1$  then
    if  $f(x_{n+1}^{t, \theta}, t) < \theta_{n+1}$  then
       $y_{n+1}^{t, \theta} = 1$ 
    else
       $y_{n+1}^{t, \theta} = 0$ 
    end if
  end if
end for
 $Loss(t, \theta, Opt) = d(t, x_N^{t, \theta})$ 

```

---

We consider a domain  $X$ , a function  $f : X \times X \rightarrow \mathbb{R}$ , and we define

$$d(t, t') = \sup_{x \in X} |f(x, t) - f(x, t')|$$

for  $t$  and  $t'$  in  $X$ . In all the paper,  $B(n, p)$  is a binomial random variable (sum of  $n$  independent Bernoulli variables of parameter  $p$ ).

The lower-bound theorem can then be stated as follows:

**Theorem 1.** *For any optimization algorithm  $Opt$ , let:*

- $N \in \mathbb{N}^*$  (a number of points visited);
- $\epsilon_0 > 0$ , and  $0 < \epsilon < \epsilon_0$  ( $\epsilon$  is a level of precision of the learned optimum);
- $D \in \mathbb{N}^*$  (characterizing a generalized dimension of  $X$ , see remark below);
- $\delta \in ]0, 1[$  (desired confidence level).

Assuming that:

- $H(\epsilon_0, D): \forall \epsilon_1 < \epsilon_0 \exists (t_1, \dots, t_D) \in X^D, \forall (i, j) \in [[1, D]]^2, i \neq j \Rightarrow d(t_i, t_j) = \epsilon_1$  (generalized dimension)
- $H_{PAC}(\epsilon, N, \delta): \forall t, P(d(x_N^{t, \theta}, t) < \epsilon/2) \geq 1 - \delta$ .

Then, if  $\delta < 1/2D$ ,

$$P(B(N, \epsilon) \geq \lceil \log_2(D) \rceil) \geq 1 - D\delta. \quad (1)$$

This theorem shows that if, with probability  $1 - \delta$ , in  $N$  iterations, we can find the solution with precision  $\epsilon$ , then Eq. 1 holds; as discussed below, this is a lower bound on the number of iterations  $N$  such that precision  $\epsilon$  is reached.

**Remarks:**

- Eq. 1 implies a clearer expression of the lower bound (using Chebychev inequality):

$$N = \Omega(\log_2(D)/\epsilon)$$

where  $N$  is the number of iterations required for reaching precision  $\epsilon$  with confidence  $1 - \delta$  for  $\delta < 1/2D$ .

- As stated above,  $H(\epsilon_0, D)$  is analogous to a dimensionality assumption, and more general. In dimension  $D$  and for simple examples of function  $f$  (see examples below), there always exists  $\Theta(D)$  points with the same distance to each other.
- $H_{PAC}(\epsilon, N, \delta)$  states that the algorithm reaches precision  $\epsilon$  with probability  $1 - \delta$  after  $N$  visited points.

**Examples.**

- A first natural example is  $X = [0, 1]^{D_0}$ ,  $\epsilon_0 = 1/10$ , and  $f(x, t) = \|x - t\|$ . In this simple cases, the theorem applies with  $D = D_0 + 1$ , and the distance  $d$  boils down to the considered norm (i.e.  $d(t, t') = \|t - t'\|$ ): reaching a precision  $\epsilon = \|x_N^\theta - t\|$  with confidence  $1 - \delta$  for any  $\delta < 1/(2D)$  will require at least  $\Omega(\log_2(D)/\epsilon)$  points.

- The fitness at the optimum needs not have a variance of 0 (consider for instance  $f(x, t) = \min(1, \frac{1}{3} + ||x - t||)$ ).<sup>1</sup>
- Interestingly, the theorem has the exact same interpretation in the less trivial case of  $f(x, t) = \min(1, ||x - t||^p)$ ,  $p > 0$ . Indeed, for  $\epsilon_0$  sufficiently small,  $H(\epsilon_0, D)$  holds for  $D = D_0 + 1$  and applying the theorem yields  $N = \Omega(\log_2(D)/\epsilon)$ . It means that, for any fixed  $p$ , the distance between the  $N^{th}$  iterate (distance for  $d(., .)$ ) and the optimum that can be guaranteed with confidence  $1 - \delta$  for  $\delta < 1/2D$ , is  $\theta(1/N)$ . Since in this case  $d(x, y) = \Theta(||x - y||)$ , a somewhat surprising conclusion can be drawn: the convergence is lower bounded by  $\Theta(1/N)$  for any value of  $p$  used in the family of fitness functions (the constant in  $\Theta$  might change, but the rate,  $N^{-1}$ , does not improve).  
Besides, for any  $p > 0$ ; there is no Lipschitz condition on the considered family of functions, and in all those cases a lower bound on  $N$  can be derived such that  $H_{PAC}(\epsilon, N, \delta)$  holds.

**Proof of Theorem 1:**

Consider a fixed  $\epsilon < \epsilon_0$ ,  $D > 0$  and  $\delta < 1/2D$ . We assume  $H(\epsilon_0, D)$  and  $H_{PAC}(\epsilon, N, \delta)$ . We consider  $t_1, \dots, t_D$  as proposed by hypothesis  $H(\epsilon_0, D)$ :

$$\forall i \neq j, d(t_i, t_j) = \epsilon. \quad (2)$$

We now show that Eq. 1 holds.

To simplify the proof, let us first introduce some relevant definition.

First, let us define

$$C_i^n(\theta) = \{j \in [[1, D]]; (y_1^{t_j, \theta}, \dots, y_n^{t_j, \theta}) = (y_1^{t_i, \theta}, \dots, y_n^{t_i, \theta})\}.$$

$C_i^n(\theta)$  is the set of  $t$  in  $t_1, \dots, t_D$  such that all answers given by the oracle are the same as on  $t_i$ . It will be useful to consider the range of values of  $f$  at the  $t_i$ 's such that  $C_i^n$  is large. For that, we define  $k_n(\theta)$  as

$$k_n(\theta) = \min\{i \in [[1, D]]; \#C_i^n(\theta) \text{ is maximal}\}.$$

The important point here is that  $k_n(\theta)$  is an index  $i$  such that  $C_i^n(\theta)$  is of maximal cardinal; the min is only here in case of equality. So,  $k_n(\theta)$  is the index of some  $t_i$  such that "many" (as many as possible)  $t_j$ 's raise the same labels  $y_1^{t_j, \theta}, \dots, y_n^{t_j, \theta}$ .

Then we define:  $s_{min, n}(\theta) = \inf_{i \in C_{k_n}^n(\theta)} f(t_i, x_n^{t_i, \theta})$ .  $s_{max, n}(\theta) = \max_{i \in C_{k_n}^n(\theta)} f(t_i, x_n^{t_i, \theta})$ .  $[s_{min, n}(\theta), s_{max, n}(\theta)]$  has the important property that it is "small"; precisely,

$$\forall(n, \theta), s_{min, n}(\theta) \geq s_{max, n}(\theta) - \epsilon \quad (3)$$

---

<sup>1</sup> Note that in our problem setting, the function  $g = \frac{1}{3} + ||x - t||$  should in all generality range in  $[0, 1]$ , which is not the case here. We can either set the measurement to 0 when  $g(x, t) > 1$ , or consider  $f(x, t) = \min(1, g(x, t))$  instead, which is what will be done from now on.

thanks to assumption  $H(\epsilon_0, D)$  and thanks to  $\epsilon < \epsilon_0$ ,  $D > 0$  and  $\delta < 1/2D$ .

The proof of Eq. 1 is now as follows:

1. Define the property  $Error(t, \theta) : d(x_N^{t, \theta}, t) \geq \epsilon/2$ . Define also  $G$ , the "good" set defined by

$$\theta \in G \text{ iff } \forall i, \#C_i^N(\theta) = 1. \quad (4)$$

Then,

$$\begin{aligned} \theta \notin G &\Rightarrow \exists i \neq j, x_{N+1}^{t_i, \theta} = x_{N+1}^{t_j, \theta} \\ &\Rightarrow Error(t_i, \theta) \text{ or } Error(t_j, \theta) \text{ thanks to Eq. 2.} \end{aligned} \quad (5)$$

2. Also,

$$H_{PAC}(\epsilon, N, \delta) \Rightarrow P(G) \geq 1 - D\delta. \quad (6)$$

**Proof of Eq. 6:**

We assume  $H_{PAC}(\epsilon, N, \delta)$ , and show that  $P(G) \geq 1 - D\delta$ .

If  $\theta \notin G$ , then  $\exists i, Error(t_i, \theta)$  (by Eq. 5 above). Therefore,

$$\begin{aligned} P(\neg G) &\leq \sum_{i \in [[1, D]]} P(Error(t_i, \theta)) \\ &\leq D\delta \text{ by } H_{PAC}(\epsilon, N, \delta). \end{aligned}$$

This concludes the proof of Eq. 6.  $\square$

3. Equation 6 and the definition of  $G$  (Eq. 4) immediately lead to (under assumption  $H_{PAC}(\epsilon, N, \delta)$ ):

$$P(\forall i, \#C_i^N(\theta) = 1) \geq 1 - D\delta. \quad (7)$$

4. We have seen with Eq. 7 that if  $H_{PAC}$  is verified, then  $C_i^N(\theta)$  must be small for all  $i$  and with high probability on  $\theta$ . We will now, in order to complete the proof, show that  $C_i^N(\theta)$  can't be small if  $N$  is not large. More precisely, we now show:

$$P(\forall i, \#C_i^N(\theta) \leq 1) \leq P(B(N, \epsilon) \geq \lceil \log_2(D) \rceil). \quad (8)$$

**Proof of Eq. 8 by induction:**

- Initialization of the induction:  $\#C_{k_0(\theta)}^0(\theta) = D$ ;
- With  $k = k_n(\theta)$  for short:
  - $P(\#C_k^n(\theta) = \#C_k^{n-1}(\theta)) \geq 1 - \epsilon$  (by Eq. 3), and
  - if  $\#C_k^n(\theta) \neq \#C_k^{n-1}(\theta)$  then  $C_k^n(\theta) \geq \lceil C_k^{n-1}(\theta)/2 \rceil$  (the set  $C_k^n\theta$  is at most divided by 2).

This implies by immediate induction that  $P(\#C_i^N(\theta) \leq 1) \leq P(B(N, \epsilon) \geq \lceil \log_2(D) \rceil)$ . This concludes the proof of Eq. 8.  $\square$

5. Equations 7 and 8 together conclude the proof of Eq. 1. This concludes the proof of Theorem 1.  $\square$

### 3 Tightness: an algorithm matching the lower bound

We have shown in the previous section a lower bound on the number of fitness evaluations for the noisy case. We now show that a bandit-based algorithm matches the bound.

We consider optimization in the domain

$$X = [x_0^-, x_0^+] = [(x_0^-)_1, (x_0^+)_1] \times [(x_0^-)_2, (x_0^+)_2] \times \cdots \times [(x_0^-)_D, (x_0^+)_D];$$

$x_0^-$  and  $x_0^+$  are elements of  $\mathbb{R}^D$  and therefore  $X \subset \mathbb{R}^D$ . We assume that  $\|x_0^- - x_0^+\| \leq 1$ .

**Sketch of Algorithm 2.** We will use Algorithm 2 for showing the tightness of Theorem 1 within logarithmic factors. It proceeds by iteratively splitting the domain in two (not necessarily equal) halves, and retaining the one that most probably contains the optimum. At iteration  $n$ , from the  $n$ th domain  $[x_n^-, x_n^+]$ , the  $(n+1)$ th domain  $[x_{n+1}^-, x_{n+1}^+]$  is obtained by:

- Finding the coordinate  $c$  such that  $\delta_n^{max} = (x_n^+)_c - (x_n^-)_c$  is maximal;
- Selecting 3 regularly spaced points along this coordinate;
- Repeatedly assessing those 3 points until we have confidence that the optimum is closer to one point  $x_n'^i$  than to another  $x_n'^j$  (by Bernstein race);
- Splitting the domain by the hyperplane in the middle of these points and normal to the line they define, and keeping only the side of the domain containing  $x_n'^i$ .

**Related works.** Our “domain reduction” step relies on a good point ( $x_n'^i$ ) and a bad point ( $x_n'^j$ ). This idea of using bad points as well as good points has already been used in [3], and in the optimisation heuristic Breda [13] (similarly to Algorithm 2, but in the deterministic case).

The remaining part of this section will be devoted to the analysis of the runtime of our algorithm. There are two points to prove regarding Algorithm 2:

- Algorithm 2 finds the optimum with precision  $\epsilon$  and confidence  $1 - \delta$ ;
- This is done with computational cost  $O(\frac{1}{\epsilon} \log(\frac{1}{\epsilon}))$ .

The following upper-bounding theorem summarizes the results of this section.

**Theorem 2 (Bandit-based noisy optimization).** *Consider a domain  $[x_0^- - x_0^+] \subset \mathbb{R}^D$  with  $\|x_0^- - x_0^+\| \leq 1$ . Consider a noisy fitness function  $f_{noise}$  with values in  $[0, 1]$  such that  $\mathbb{E}f_{noise}(x) = \|x - x^*\|$  and  $x^* \in [x_0^-, x_0^+]$ .<sup>2</sup> Then, Algorithm 2 ensures the following properties:*

- **Soundness:** *The  $n^{th}$  domain has diameter  $\leq \sqrt{D}(\frac{3}{4})^{\lfloor n/D \rfloor}$ ;*
- **Consistency:** *With probability at least  $1 - \delta$ , for all  $n > 0$  the optimum is in the  $n^{th}$  domain  $[x_n^-, x_n^+]$ ;*
- **Rate:** *For a fixed  $D$ , the number of function evaluations before the domain has diameter  $\leq \epsilon$  is  $O(\frac{1}{\epsilon} \log(1/\epsilon))$ .*

<sup>2</sup> Note that this formulation is equivalent to the problem setting described at the beginning of section 2, with  $x^*$  in the role of  $t$ ; it is more suited to describe our algorithm and proof.



*Remark: the multi-armed bandit framework.* This theorem is termed “bandit-based” because it is inspired from the multi-armed bandit framework (see e.g. [4]). The multi-armed bandit framework is named after the famous “one-armed bandit” machines. There are multiple *arms* that can be tried by a player, each giving a stochastic reward with unknown mean and variance. The goal is to play the machine that yields the best mean reward most of the time.

In this paper, points of the domain  $X$  are similar to arms: they yield a random “reward” (between 0 and 1) and the goal is to find the one that has the best mean (the optimum). This is the spirit of the Bernstein Race algorithm, Algorithm 3.

*Remark: Bernstein races.* Hoeffding’s bound [18] is a classical concentration inequality: it shows that the difference between the averages and the expectations in “small”, if the sample size is enough. It can be applied to so-called Hoeffding’s races[22, 23]: given a sample of candidates, one might evaluate all of them, get the stochastic rewards, and keep only for further evaluations the candidates for which the Hoeffding’s bound is not too bad. After several iterations, only good candidates are kept.

Bernstein’s bound [10, 6, 7] is an improvement of Hoeffding’s bound which takes into account the variance of candidates, and not only their range. There are variants of Bernstein’s bounds which can even take into account the empirical variance, and not only the variance itself; these bounds can therefore directly be used instead of Hoeffding’s bound in races—this is Bernstein races. The results about Bernstein races used in this paper can be found in [24] and [11, p124].

We first give an intuitive idea of our proof, and then the complete proof of Theorem 2.**Proof sketch.** The algorithm geometrically decreases the domain, hence the soundness property. Thanks to the specific family of functions (the sphere), the optimum is closer to the “good” arm than to the “bad arm” with high probability: this provides the consistency. The number of requests to the fitness functions will then be upper bounded by classical Bernstein bounds; this is only possible thanks to the specific mutation operator in Algorithm 2: it ensures that arms are all “sufficiently different”.

### **Proof of Theorem 2:**

In the following,  $\mathbb{E}f_{noise}(x)$  will be referred to as  $f(x)$ , and  $\hat{f}(x)$  will stand for an empirical estimate of  $f(x)$  by averaging the rewards given by  $f_{noise}(x)$ .

First of all, notice that  $f_{noise}(x) \in [0, 1]$  implies  $Var f_{noise}(x) \leq E[f_{noise}(x)] = ||x - x^*||$ . A (quick and easy) proof of this is provided in appendix.

A step towards proving theorem 2 is stated by the following lemma:

---

**Algorithm 2** Algorithm for optimizing noisy fitness functions. *Bernstein* denotes a Bernstein race, as defined in Algorithm 3. The algorithm is illustrated in Fig. 1.

---

```

 $n \leftarrow 0$ 
 $K_0 = \sqrt{1+D} - \sqrt{\frac{1}{4}+D}$ 
while True do
   $c = \arg \max_i (x_n^+)_i - (x_n^-)_i$  // we pick up the coordinate with highest uncertainty
   $\delta_n^{\max} = (x_n^+)_c - (x_n^-)_c$ 
  for  $i \in [[1, 3]]$  do
     $x_n'^i = \frac{1}{2}(x_n^- + x_n^+)$ . // we consider the middle point
     $(x_n'^i)_c = (x_n^-)_c + \frac{i-1}{2}(x_n^+ - x_n^-)_c$ . // except that we modify the  $c^{th}$  coordinate
  end for
   $\epsilon_1 = \sqrt{D}\delta_n^{\max}$ . // this is a bound on the expected fitness of the  $x_n'^i$ 
   $\epsilon_2 = 2K_0\delta_n^{\max}$ . // this is a lower bound on the difference between
  // the best and the worst arm
   $(good, bad) = \text{Bernstein}(x_n'^1, x_n'^2, x_n'^3, \frac{6\delta}{\pi^2(n+1)^2}, \epsilon_1, \epsilon_2)$ .
  Let  $H$  be the halfspace  $\{x \in \mathbb{R}^D; \|x - good\| \leq \|x - bad\|\}$ .
  Split the domain:  $[x_{n+1}^-, x_{n+1}^+] = H \cap [x_n^-, x_n^+]$ .
   $n \leftarrow n + 1$ 
end while

```

---

**Lemma 1 (Convergence of Alg. 2).** *If Algorithm 2 reaches the  $n_0^{th}$  iteration and  $x^*$  is in the  $n_0^{th}$  domain, then for all  $n \leq n_0$ ,  $\delta_n = x_n^+ - x_n^-$  verifies:*

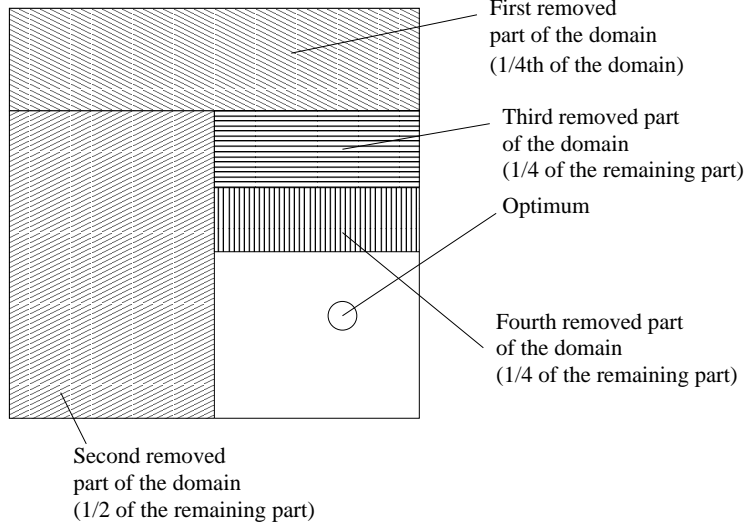
$$\begin{aligned}
\delta_n^c &\leq \delta_{n-1}^c \times \frac{3}{4} \text{ where } c = \arg \max_i (x_n^+)_i - (x_n^-)_i; \\
\delta_{n+D}^{\max} &\leq \delta_n^{\max} \times \frac{3}{4} \text{ where } \delta_i^{\max} = \sup_j (\delta_i)_j; \\
\delta_n^{\max} &\leq \left(\frac{3}{4}\right)^{\lfloor n/D \rfloor}.
\end{aligned}$$

**Remark:** Since it hasn't yet been proved that Algorithm 3 terminates with  $x^*$  in the remaining domain, the result is stated under those conditions. It will be shown in propositions 1 and 2 that those conditions are satisfied with high probability.

**Proof of Lemma 1:**

This is an immediate property of Algorithm 2: at each iteration, the part of the domain which is on the wrong side of  $H$  (meaning that  $x^*$  is in the other part, see Fig. 2) is removed, and its size is at least  $\delta_n^{\max}/4$ .  $\square$

We will now prove that provided Algorithm 2 reaches iteration  $n$ , the assumptions made in Algorithm 3 guaranteeing its proper behavior are met.



**Fig. 1.** Noisy optimization algorithm (cf Algorithm 2). At each iteration, a main axis is selected (the one on which the domain has maximum range). Three equally spaced points are generated in the domain on this axis (this is the offspring). Then, a Bernstein race is applied for choosing a “good” and a “bad” arm among these points. The domain is reduced thanks to this knowledge, removing one fourth or one half of the domain (depending on the position of the good arm and of the bad arm - the best case is when the good and the bad arm are diametrically opposed: see Fig. 2).

**Lemma 2 (The conditions of the Bernstein race are met).** *Assume that  $x^* \in [x_n^-, x_n^+]$ . Then*

$$\max_{(i,j) \in [[1,3]]^2} f(x_n'^j) - f(x_n'^i) \geq 2K_0 \delta_{max}^n$$

$$\text{where } K_0 = \left( \frac{1}{4} \sqrt{1+D} - \sqrt{\frac{1}{4} + D} \right); \quad (10)$$

$$\max_{i \in [[1,3]]} f(x_n'^i) \leq \sqrt{D} \delta_{max}^n. \quad (11)$$

**Proof of Lemma 2:**

Eq. 11 is immediate by definition of  $f$ . Let us show why eq. 10 holds.

Let  $\bar{x}_n^*$  be the projection of  $x^*$  on the line on which lie  $x_n'^1, x_n'^2, x_n'^3$ . The result will now be proved for  $(\bar{x}_n^*)_c \in [(x_n'^1)_c, (x_n'^2)_c]$ . The proof for the case  $(\bar{x}_n^*)_c \in [(x_n'^2)_c, (x_n'^3)_c]$  is symmetric (see Figure 2).

First of all, we have

$$\Delta_n \doteq \max_{i,j \in [[1,3]]^2} f(x_n'^i) - f(x_n'^j) \geq f(x_n'^3) - f(x_n'^2)$$

---

**Algorithm 3** Bernstein comparison between 3 arms. Eq. 9 is Bernstein's inequality for estimating the risk, for arms with variance  $\leq \epsilon_1$  and precision  $\epsilon$  (see e.g. [11, p124]).

---

*Bernstein*( $a_1, a_2, a_3, \delta', \epsilon_1, \epsilon_2$ )  
 //We assume that arm  $a_1$  (resp.  $a_2, a_3$ ) has (expected) fitness  $p_1$  (resp.  $p_2, p_3$ ).  
 //We assume that  $p_1, p_2, p_3 \geq 0$ .  
 //We assume that  $\epsilon_1 \geq \max(p_1, p_2, p_3)$ .  
 //We assume that  $\epsilon_2 \leq \max_i p_i - \min_i p_i$ .  
 $t = 0, \epsilon = \epsilon_2/2$   
**repeat**  
    $t \leftarrow t + 1$   
   Evaluate the fitness of arms  $a_1, a_2, a_3$  once.  
   Evaluate the precision:

$$\epsilon_{(t)} = \sup\{\epsilon > 0; 3 \exp\left(-\frac{t\epsilon^2}{2\epsilon_1 + 2\epsilon/3}\right) \leq \frac{6\delta'}{\pi^2 t^2}\}. \quad (9)$$

**until** There are two arms *good* and *bad* with empirical difference  $\geq 2\epsilon$ :  $\hat{f}(\text{good}) - \hat{f}(\text{bad}) \geq 2\epsilon$

---

By Pythagora's theorem,  $\forall i \in [1, 3], \|x_n'^i - x^*\|^2 = \|x_n'^i - \bar{x}_n^*\|^2 - \|\bar{x}_n^* - x^*\|^2$ . Thus,

$$\Delta_n \geq \sqrt{\|x_n'^3 - \bar{x}_n^*\|^2 - \|\bar{x}_n^* - x^*\|^2} - \sqrt{\|x_n'^2 - \bar{x}_n^*\|^2 - \|\bar{x}_n^* - x^*\|^2}$$

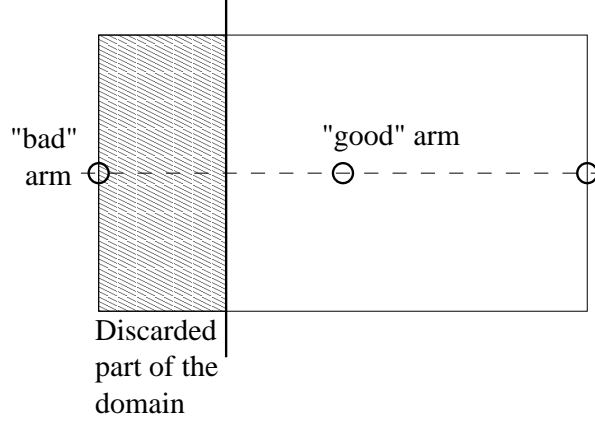
Note that  $\|x_n'^3 - \bar{x}_n^*\| = \|x_n'^2 - \bar{x}_n^*\| + \delta_n^{\max}/2$ . Define  $d = \|\bar{x}_n^* - x^*\|^2$  and  $a = \|x_n'^3 - \bar{x}_n^*\|$ . Then, observing that  $\delta_n^{\max} \geq a \geq \delta_n^{\max}/2$ , we have

$$\begin{aligned} \Delta_n &\geq \sqrt{a^2 + d} - \sqrt{(a - \delta_n^{\max}/2)^2 + d} \\ &\geq a \left( \sqrt{1 + d/a^2} - \sqrt{\left(1 - \frac{\delta_n^{\max}}{2a}\right)^2 + \frac{d}{a^2}} \right) \\ &\geq \frac{\delta_n^{\max}}{2} \left( \sqrt{1 + d/a^2} - \sqrt{\frac{1}{4} + \frac{d}{a^2}} \right) \end{aligned} \quad (12)$$

By setting  $u = d/a^2$ , it is clear that  $\Delta_n$  is greater than the minimum of  $u \mapsto \sqrt{1+u} - \sqrt{1/4+u}$  on the interval  $[0, D]$  (since  $\sqrt{d} = \|\bar{x}_n^* - x^*\| \leq \sqrt{D}\delta_n^{\max}/2$ ). It remains to find the minimum of that function. Its derivative is

$$\frac{1}{2\sqrt{1+u}} - \frac{1}{2\sqrt{1/4+u}}$$

which is always negative, since the left-hand side denominator is always bigger than the right-hand side denominator. Therefore, the function is decreasing, and is minimized for  $u = \sqrt{D}$ . Finally, setting  $K_0 = 1/4(\sqrt{1+D} - \sqrt{1/4+D})$  and injecting in eq. 12 yields  $\Delta_n \geq 2K_0\delta_n^{\max}$ , as stated by Eq. 10.  $\square$



**Fig. 2.** The large rectangle is the domain  $[x_n^-, x_n^+]$ . The three circles are arms  $x_n'^1, x_n'^2, x_n'^3$ ; the left arm is the “bad” arm, whereas the arm in the center is the “good” arm, *i.e.* the one which proved to be closer to the optimum than the left arm, with confidence  $1 - 6\delta/(\pi^2 n^2)$ .

Lemma 1 shows that the domain gets smaller and smaller at a given fixed rate. Lemma 2 shows that, if the optimum is in  $[x_n^-, x_n^+]$ , then the conditions of Algorithm 3, as called in Algorithm 2, are met, *i.e.* all arms have a probability  $\leq \epsilon_1$  and there are two arms with parameters  $p_i$  and  $p_j$  such that  $p_i > p_j + \epsilon_2$ .

It implies the following proposition:

**Proposition 1 (Consistency of Algorithm 3).** *In the previously defined setting in which algorithms 2 and 3 operate, with probability at least  $1 - \delta$ , for all calls to Algorithm 3:*

$$\begin{aligned} \forall n, \text{ in the call to Alg. 3 in iteration } n \text{ of Alg. 2,} \\ \forall t, \forall i \in \{1, 2, 3\}, |\hat{p}_i - p_i| \leq \epsilon_{(t)} \end{aligned} \quad (13)$$

and therefore

$$\forall n, x^* \in [x_n^-, x_n^+]. \quad (14)$$

$$\text{and } \delta_n^{\max} \leq \left(\frac{4}{5}\right)^{\lfloor n/D \rfloor} \quad (15)$$

**Proof of Proposition 1:**

The fact that  $\delta_n^{\max} \leq \left(\frac{4}{5}\right)^{\lfloor n/D \rfloor}$  is already proved in Lemma 1. We have to show Eq. 13 and Eq. 14.

This can be shown by induction on  $n$ .

(i) The case  $n = 0$  holds. This is because in our setting, it is stated that  $x^* \in [x_0^-, x_0^+]$  and  $\delta_{\max}^0 < 1$ ; this shows Eq. 14 for  $n = 0$ . By Lemma 2, we can apply Bernstein’s race (see e.g. [11, p124] or [24]), which state Eq. 13. Equations 13 and 14 are therefore proved for  $n = 0$ .

(ii) Let's assume Equations 13 and 14 for a given  $n$ , and let's prove them for  $n + 1$ . Eq. 14 and the halting condition of Alg. 3 ensure that Algorithm 3 terminates and is right (and, by definition of the Bernstein race, it will terminate and be right with probability at least  $1 - \delta'$ ), its output is 2 arms ( $good, bad$ ) such that  $f(good) < f(bad)$ .

Thus, the split that occurs in Algorithm 2 after the call to the Bernstein Race ensures with probability  $1 - \delta'$  that  $x^*$  remains in the  $(n+1)$ th domain; this is Eq. 14. Besides, Lemma 1 guarantees that the domain's edge of length  $\delta_n^{\max}$  decreases as wished (Eq. 15). We have now to show Eq. 13. From Eq. 14, Lemma 2 states that the conditions of the Bernstein race are met; This implies Eq. 13 by usual properties of Bernstein races.

(iii) Each induction step  $n$  holds with probability  $1 - \delta'_n$  if the preceeding one holds, so the proposition is true for all  $n$  with probability at least

$$\sum_{n=0}^{\infty} (1 - \delta'_n) = \sum_{n=0}^{\infty} \left(1 - \frac{6\delta}{\pi^2(n+1)^2}\right) = \delta$$

which concludes the proof.  $\square$

It remains to prove that, at each iteration, Algorithm 3 requires a finite number of requests to the fitness function small enough so that in the end the lower bound is reached:

**Proposition 2 (Number of iterations of the Bernstein race).** *Consider Algorithm 3 called by Algorithm 2, for a given iteration  $n$  with*

$$x^* \in [x_n^-, x_n^+] \quad (16)$$

*Assume that*

$$\forall t, \forall i \in \{1, 2, 3\}, |\hat{p}_i - p_i| \leq \epsilon_{(t)} \quad (17)$$

*Then, with probability  $1 - \delta'$ , the number of iterations  $t$  before the halting condition in Algorithm 3 is  $O(-\log(\epsilon_1)/\epsilon_1)$ .*

**Remarks.** We have assumed Eq. 17 and Eq. 16. These two assumptions are verified, for all  $n$ , with probability at least  $1 - \delta$ , as shown in Prop. 1; we can therefore apply Proposition 1 and Proposition 2 simultaneously, and for all  $n$ , with probability  $1 - \delta$ .

**Proof of Proposition 2:**

Let's bound the number of fitness evaluations of a call to the Bernstein race. We consider a fixed  $D$ , and we will use the following facts:

- Algorithm 3 is called with  $\epsilon_2 = \Theta(\epsilon_1)$ ;
- The iteration  $n$  of Algorithm 2 verifies  $n = \Theta(\log(\epsilon_1))$ ;
- The Bernstein race 3 stops as soon as  $\exists i, j; \epsilon_{(t)} \leq \frac{1}{4}(p_i - p_j)$ , and therefore necessarily the last iteration  $t_n$  of Algorithm 3 when called at iteration  $n$  of algorithm 2 is such that

$$\epsilon_{(t_n-1)} > \epsilon_2/4 \quad (18)$$

Plugging Eq. 18 in Eq. 9 and using  $\epsilon_2 = \Theta(\epsilon_1)$ ,  $n = \Theta(\log(\epsilon_1))$ ,  $\delta' = \theta(1/n^2)$ , yields

$$\exp(-t_n \Theta(\epsilon_1)) \geq \Theta(1/n^2 t^2). \quad (19)$$

Eq. 19 yields

$$t_n = O(-\log(\epsilon_1)/\epsilon_1). \quad (20)$$

This concludes the proof of Proposition 2.  $\square$

We have, with Eq. 20, a bound on the number of fitness evaluations in terms of  $\epsilon_1$ . Consider a precision  $\epsilon$ . If  $x^* \in [x_N^-, x_N^+]$  and  $\sqrt{D}\delta_N^{max} \leq \epsilon$ , then necessarily  $\|x_N^- x^*\| \leq \epsilon$ ; therefore, we have reached precision  $\epsilon$  if  $\delta_N^{max} \leq \epsilon/\sqrt{D}$  and the consistency holds. Consider the case in which consistency holds (we have proved, in Proposition 1, that this holds with probability  $1 - \delta$ ).

Then, if  $n$  is the last iteration in which the Bernstein race is applied,  $\delta_n^{max} \geq \epsilon/\sqrt{D}$ . This equation, combined with Eq. 20, gives an upper bound  $L(\epsilon)$  as follows on the complexity (in terms of the number of fitness evaluations) of the last call to *Bernstein*(.):

$$L(\epsilon) = O(-\log(\epsilon)/\epsilon).$$

We now have to bound the complexity of the whole run of Algorithm 2, and not only the last call to the Bernstein race; we'll see that the cost is indeed essentially in the last iteration. The complexity of the last  $D$  iterations (iterations  $N, N-1, \dots, N+1-D$ ) are also bounded by  $L(\epsilon)$ , because  $\delta_{max}^n$  is non-increasing and our bound (Eq. 20) on the cost of one iteration is decreasing as a function of  $\epsilon_1 = \sqrt{D}\delta_n^{max}$ . Therefore, the cost for these  $D$  iterations is at most  $DL(\epsilon)$ . Each of the  $D$  previous iterations (iterations  $N-D, N-1-D, \dots, N+1-2D$ ) has computational cost bounded by  $(\frac{3}{4})L(\epsilon)$ , because

- for  $n' \leq n - D$ ,  $\delta_{max}^{n'} \geq \frac{4}{3}\epsilon$ ;
- the cost (Eq. 20) is superlinear as a function of  $\epsilon_1 = \sqrt{D}\delta_{max}^n$ .

In the same way, the cost of iteration  $N - kd - i$ , for some  $k \in \mathbb{N}$  and  $i \in [[0, D-1]]$ , is bounded from above by  $L(\epsilon)(\frac{3}{4})^k$  if  $N - kd - i \geq K_2(D)$ .

The overall cost is therefore bounded from above by

$$DL(\epsilon) \times (1 + \frac{3}{4} + (\frac{3}{4})^2 + (\frac{3}{4})^3 + \dots).$$

Therefore the computational cost for ensuring

$$\sqrt{D}\delta_N^{max} \leq \epsilon$$

is  $O(DL(\epsilon))$ , and is therefore  $O(-\log(\epsilon)/\epsilon)$ . The tightness within logarithmic factors is proved, and Theorem 2 is proved.  $\square$

*Further remarks:*

- We have not optimized the dependency in  $D$ . Our algorithm does not need that the fitness values at a given point is Bernoulli; it works for any distribution with values between 0 and 1 and variance upper bounded by the expectation.
- Our algorithm uses a specific form of Bernstein race, in which  $\epsilon_1$  and  $\epsilon_2$  are known in advance (i.e. from  $\delta_n^{max}$  we can guess an upper bound on the fitness values in  $[x_n^-, x_n^+]$  and a lower bound on the difference between the best and the worst arm). It means that the algorithm works e.g. for  $f_{noise}(x)$  a Bernoulli random variable  $B(\|x - x^*\|)$  with parameter  $\|x - x^*\|$ , but not for  $f_{noise}(x) = B(2\|x - x^*\|)$  or  $f_{noise}(x) = B(k\|x - x^*\|)$  for arbitrary  $k > 0$ . Fortunately, [24] provides other Bernstein-based races, bearing no significant increase of the computational cost without this prior knowledge; therefore, replacing our simple Bernstein race by one of these results in the same soundness, and the same consistency; the convergence rate is only modified in the sense that it holds for a fixed  $D$  and a fixed  $k$ . This is a simple modification; a less trivial further work consists in analyzing this algorithm with a wider family of fitness functions; we guess that the soundness and consistency are preserved for any fitness function with values in  $[0, 1]$  with  $\mathbb{E}f_{noise} = g(\|x - x^*\|)$  with  $g$  some increasing function from  $[0, \infty[$  to  $[0, 1]$ , and optimal complexity (within logarithmic factors)  $O(1/\epsilon^2)$  for e.g.  $g(t) = c + t$  for some  $c > 0$ .

## 4 Conclusion

The key result of this paper is that including a bandit as in [17] in the evaluation step of an evolution strategy is a good idea. The underlying subtlety is that we must not find all the  $\mu$  best points as in [17]; when trying to find all the  $\mu$  best points (even if  $\mu = 1$ !) one might spend of huge computational effort for separating two points with very close fitness. This happened with several tested variants. It is much more stable to generate an offspring in a controlled manner—so that there’s no risk of having all points with almost equal fitness—and then to try and ensure that one of the points is better than another one, nothing more.

**Further work.** A first extension to the results presented in this paper consists in generalizing the result for including “natural” algorithms; our algorithm is somehow reasonable, but implies specifically derandomized offspring; practitioners might prefer more classical mutation algorithms, with an ad hoc bandit algorithm for avoiding troubles.

Another further work has been outlined after Theorem 2: considering a Bernstein race without prior on the shape of bandits, with empirical variance estimates instead. This is easy to do in our setting, but a non trivial extension would be the use for cases in which the variance of the noise around the optimum does not converge to zero.



We have shown a lower bound on the complexity of optimization of noisy fitness functions, with assumptions on the variance of noise around the optimum. There is no log-linear convergence (for convergence rate normalized by the number of function evaluations), but instead a linear decrease of the precision as a function of the number of iterations. We guess that the lower bound becomes the square root of the inverse of the number of evaluations if the variance does not decrease to zero around the optimum—proving it is another necessary further work.

Parallelization should provide a much better speed-up in the case of noisy optimization than in deterministic frameworks. It is therefore quite appealing to extend the above results for more than 1 computation unit.

## A

We will show that given a random variable  $X$  whose probability distribution's support is in  $[0,1]$ , its variance is always bounded by its mean. Let us assume that the distribution admits a density  $p$ . Then, define  $\alpha = E[X] = \int_0^1 xp(x)dx$ .

$$\begin{aligned} E[X] - E[(X - E[X])^2] &= \int_0^1 (x - (x - \alpha)^2)p(x)dx \\ &= (1 + 2\alpha)\alpha - \alpha^2 - \int_0^1 x^2p(x)dx \\ &= \alpha^2 + \int_0^1 (x - x^2)p(x)dx \end{aligned}$$

Since  $\forall x \in ]0,1[, x > x^2$  and  $p(x) \geq 0$ ,  $E[X] - E[(X - E[X])^2]$  is positive, which proves the original claim for variables admitting a density. Furthermore, it is clear that the result is also true for random variables who do not admit a density, by seeing their probability distribution as a limit of a sequence of probability distributions admitting densities, and applying the above reasoning.

**Acknowledgements** This work was supported by the French National Research Agency (ANR) grant No. ANR-08-COSI-007-12, and through COSINUS program (project EXPLO-RA nANR-08-COSI-004). We are very grateful to Hervé Fournier for his valuable comments on a preliminary version of this paper, and to Anne Auger for fruitful discussions around noisy optimization. We are also grateful to the people of the “MoGo” project for joint work around noisy optimization.

## References

1. D. V. Arnold and H.-G. Beyer. Efficiency and mutation strength adaptation of the (mu/mui,lambda)-es in a noisy environment. In M. S. et al., editor, *Parallel Problem Solving from Nature*, volume 1917 of *LNCS*, pages 39–48. Springer, 2000.

2. D. V. Arnold and H. Georg Beyer. Evolution strategies with cumulative step length adaptation on the noisy parabolic ridge. Technical report, 2006.
3. D. V. Arnold and D. C. S. V. Wart. Cumulative step length adaptation for evolution strategies using negative recombination weights. In *EvoWorkshops*, pages 545–554, 2008.
4. P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2/3):235–256, 2002.
5. A. Auger. Convergence results for  $(1,\lambda)$ -SA-ES using the theory of  $\varphi$ -irreducible markov chains. *Theoretical Computer Science*, 334:35–69, 2005.
6. S. Bernstein. On a modification of chebyshev’s inequality and of the error formula of laplace. *Original publication: Ann. Sci. Inst. Sav. Ukraine, Sect. Math.* 1, 3(1):38–49, 1924.
7. S. Bernstein. *The Theory of Probabilities*. Gastehizdat Publishing House, Moscow, 1946.
8. H.-G. Beyer. *The Theory of Evolutions Strategies*. Springer, Heidelberg, 2001.
9. A. Bienvenue and O. Francois. Global convergence for evolution strategies in spherical problems: some simple proofs and difficulties. *Theor. Comput. Sci.*, 306(1-3):269–289, 2003.
10. H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Math. Stat.*, 23:493–509, 1952.
11. L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic Theory of Pattern Recognition*. Springer, 1997.
12. J. M. Fitzpatrick and J. J. Grefenstette. Genetic algorithms in noisy environments. *Machine Learning*, 3:101–120, 1988.
13. S. Gelly, S. Ruetten, and O. Teytaud. Comparison-based algorithms are robust and randomized algorithms are anytime. *Evolutionary Computation*, 15(4):411–434, 2007.
14. U. Hammel and T. Bäck. Evolution strategies on noisy functions: How to improve convergence properties. In Y. Davidor, H.-P. Schwefel, and R. Manner, editors, *Parallel Problem Solving From Nature*, volume 866 of *LNCS*, pages 159–168, Jerusalem, 9–14Oct. 1994. Springer.
15. N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 11(1), 2003.
16. V. Heidrich-Meisner and C. Igel. Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. In *ICML ’09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 401–408, New York, NY, USA, 2009. ACM.
17. V. Heidrich-Meisner and C. Igel. Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. In *ICML ’09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 401–408, New York, NY, USA, 2009. ACM.
18. W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.
19. M. Jebalia and A. Auger. On multiplicative noise models for stochastic search. In *Parallel Problem Solving From Nature*, dortmund Allemagne, 2008.
20. M. Jebalia and A. Auger. On multiplicative noise models for stochastic search. In *Parallel Problem Solving From Nature*, dortmund Allemagne, 2008.
21. T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6:4–22, 1985.

22. O. Maron and A. W. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In *In Advances in neural information processing systems 6*, pages 59–66. Morgan Kaufmann, 1994.
23. O. Maron and A. W. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11:193–225, 1997.
24. V. Mnih, C. Szepesvári, and J.-Y. Audibert. Empirical Bernstein stopping. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 672–679, New York, NY, USA, 2008. ACM.
25. I. Rechenberg. *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien des Biologischen Evolution*. Fromman-Holzboog Verlag, Stuttgart, 1973.
26. H.-P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, New-York, 1981. 1995 – 2<sup>nd</sup> edition.
27. O. Teytaud and A. Auger. On the adaptation of the noise level for stochastic optimization. In *IEEE Congress on Evolutionary Computation*, Singapour, 2007.